APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/010,389	11/08/2001	Matthew Becker	SMQ-143/P6594	4465

959   7590   07/15/2004

LAHIVE & COCKFIELD, LLP.
28 STATE STREET
BOSTON, MA   02109

EXAMINER
RAVINDRAN, LATHA

ART UNIT	PAPER NUMBER
2183	

DATE MAILED: 07/15/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 10/03)

Office Action Summary	Application No.	Applicant(s)
	10/010,389	BECKER ET AL.
	Examiner	Art Unit
	Latha Ravindran	2183

-- *The MAILING DATE of this communication appears on the cover sheet with the correspondence address* --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.
- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *08 November 2001*.

2a)☐ This action is **FINAL**.   2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-17* is/are pending in the application.

     4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-17* is/are rejected.

7)☒ Claim(s) *1-17* is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *08 November 2001* is/are: a)☐ accepted or b)☒ objected to by the Examiner.

     Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

     Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

     a)☐ All   b)☐ Some * c)☐ None of:

        1.☐ Certified copies of the priority documents have been received.

        2.☐ Certified copies of the priority documents have been received in Application No. _____.

        3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

     * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1)☒ Notice of References Cited (PTO-892)

2)☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3)☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date *06192002*.

4)☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

5)☐ Notice of Informal Patent Application (PTO-152)

6)☐ Other: _____.

DETAILED ACTION

1. Claims 1 - 17 have been examined. Claims 1 – 17 have been rejected.

2. Receipt is acknowledged for Declaration, received 2/11/2002, Information
Disclosure Statement, received 6/19/2002, and Change in Power of Attorney,
received 4/26/2004.

Drawings

3. The drawings are objected to under 37 CFR 1.83(a). The drawings must show
every feature of the invention specified in the claims. Therefore, the following
must be shown or the feature(s) canceled from the claim(s). No new matter
should be entered.

Claim 1:

- advancing instructions along a microprocessor pipeline (fig. 3 and fig. 4 are block
diagrams of the pipeline stages)

- edge detecting valid instructions within the microprocessor pipeline

Claim 2:

- edge detecting valid instructions within the bundle

Claim 4:

- rotating at least one instruction based at least in part on the number of valid
instructions in the bundle

Claim 5:

- compressing the bundle of instructions

Claim 6:

- compressing the bundle of instructions for a monotonic instruction set

Claim 7:

- compressing the bundle of instructions based at least in part on the number of valid instructions in the bundle

Claim 8:

- edge detecting the number of valid instructions occurring after the complex instruction.

Claim 12:

- the step of shifting the instructions comprises compressing the instructions occurring after the complex instruction

Claim 13:

- the step of shifting the instructions comprises compressing the instructions occurring after the complex instruction for a monotonic instruction set

Claim 14:

- executing instructions occurring prior to the complex instruction during a first clock cycle

Claim 15:

- executing the complex instruction during a second clock cycle

Claim 16:

- the step of shifting the instructions occurs while at least one of i) the instructions occurring prior to the complex instruction are executed and ii) the complex instruction is executed.

Claim 17:

- executing during a first clock cycle valid instructions occurring prior to the complex instruction;

- executing the complex instruction during a second clock cycle;

- shifting instructions occurring after the complex instruction during at least one of the first clock cycle and the second clock cycle;

- edge detecting valid instructions occurring after the complex instruction during at least one of the first clock cycle and the second clock cycle; and

- executing the valid instructions occurring after the complex instruction during a third clock cycle.

4. Corrected drawing sheets are required in reply to the Office action to avoid abandonment of the application. Any amended replacement drawing sheet should include all of the figures appearing on the immediate prior version of the sheet, even if only one figure is being amended. The figure or figure number of an amended drawing should not be labeled as "amended." If a drawing figure is to be canceled, the appropriate figure must be removed from the replacement sheet, and where necessary, the remaining figures must be renumbered and appropriate changes made to the brief description of the several views of the drawings for consistency. Additional replacement sheets may be necessary to show the renumbering of the remaining figures.

5. Figures 1,2,3,4, and 5 should be designated by a legend such as --Prior Art-- because only that which is old is illustrated. See MPEP § 608.02(g). Corrected

drawing sheets are required in reply to the Office action to avoid abandonment of

the application.

6. The replacement sheet(s) should be labeled "Replacement Sheet" in the page

   header (as per 37 CFR 1.84(c)) so as not to obstruct any portion of the drawing

   figures. If the changes are not accepted by the examiner, the applicant will be

   notified and informed of any required corrective action in the next Office action.

   The objections to the drawings will not be held in abeyance.

Specification

7. Applicant is reminded of the proper language and format for an abstract of the

   disclosure.

   The language should be clear and concise and should not repeat information
   given in the title. It should avoid using phrases which can be implied, such as,
   "The disclosure concerns," "The disclosure defined by this invention," "The
   disclosure describes," etc.

8. Remove or correct the first sentence of the abstract because it repeats the title.

9. A new title is required that is clearly indicative of the invention to which the claims

   are directed. There is no "system" being claimed. Remove the reference to

   system out of the title.

10. The following title is suggested: Methods for Determining Valid Microprocessor

    Instructions in an Instruction Bundle.

11. Remove the reference to systems in the "Field of Invention" because only

    methods are described.

12. The disclosure is objected to because it contains an embedded hyperlink and/or

    other form of browser-executable code. (Page 4, paragraph 13) Applicant is

required to delete the embedded hyperlink and/or other form of browser-executable code. See MPEP § 608.01.

13. The specification is objected to as failing to provide proper antecedent basis for the claimed subject matter. See 37 CFR 1.75(d)(1) and MPEP § 608.01(o). Correction of the following is required:.

Claim 4:

There is no mention of "rotating at least one instruction based at least in part on the number of valid instructions in the bundle" in the written description. One of ordinary skill in the art would not recognize the written description provides support for the claim because there is no explicit mention of this step in the written description, and the preferred embodiment does not inherently include this step. Instead the preferred embodiment shifts the instruction in the bundle. What does it mean to rotate at least one instruction? The examiner notes, that upon further examination, rotate is interpreted as, "A variation of a logical shift in which the digits moved out of one end of a register, word, or numeral are returned to the other end." (definition of circular shift, The IEEE Standard Dictionary of Electrical and Electronics Terms, Sixth Edition)

Claim 5:

There is no mention of "compressing the bundle of instructions" in the written description. One of ordinary skill in the art would not recognize the written description provides support for the claim because there is no explicit mention of this step in the written description, and the preferred embodiment does not

inherently include this step. The examiner notes, that upon further examination, compress is interpreted as, " to reduce the size of a set of data, such as a file or a communications message, so that it can be stored in less space or transmitted with less bandwidth." (Microsoft Computer Dictionary, Third Edition)

Claim 6:

There is no mention of "compressing the bundle of instructions for a monotonic instruction set" in the written description. One of ordinary skill in the art would not recognize the written description provides support for the claim because there is no explicit mention of this step in the written description, and the preferred embodiment does not inherently include this step. The examiner notes, that upon further examination, compress is interpreted as, " to reduce the size of a set of data, such as a file or a communications message, so that it can be stored in less space or transmitted with less bandwidth." (Microsoft Computer Dictionary, Third Edition)

Claim 7:

There is no mention of "compressing the bundle of instructions based at least in part on the number of valid instructions in the bundle" in the written description. One of ordinary skill in the art would not recognize the written description provides support for the claim because there is no explicit mention of this step in the written description, and the preferred embodiment does not inherently include this step. The examiner notes, that upon further examination, compress is interpreted as, " to reduce the size of a set of data, such as a file or a

communications message, so that it can be stored in less space or transmitted

with less bandwidth." (Microsoft Computer Dictionary, Third Edition)

Claim 9:

The term "bundling" lacks antecedent basis in the description. While instruction

bundle is defined in the description, the verb bundling is not. What does bundling

entail? The examiner notes, that upon further examination, bundle (the verb) is

interpreted as, "to tie, wrap, or gather together." (Webster's II New Riverside

University Dictionary)

Claim 11:

The term "bundling" lacks antecedent basis in the description. While instruction

bundle is defined in the description, the verb bundling is not. What does bundling

entail? The examiner notes, that upon further examination, bundle (the verb) is

interpreted as, "to tie, wrap, or gather together." (Webster's II New Riverside

University Dictionary)

Claim 12:

There is no mention of "the step of shifting the instructions comprises

compressing the instructions occurring after the complex instruction" in the

written description. One of ordinary skill in the art would not recognize the written

description provides support for the claim because there is no explicit mention of

this step in the written description, and the preferred embodiment does not

inherently include this step. The examiner notes, that upon further examination,

compress is interpreted as, " to reduce the size of a set of data, such as a file or

a communications message, so that it can be stored in less space or transmitted with less bandwidth." (Microsoft Computer Dictionary, Third Edition)

Claim 13:

There is no mention of "the step of shifting the instructions comprises compressing the instructions occurring after the complex instruction for a monotonic instruction set" One of ordinary skill in the art would not recognize the written description provides support for the claim because there is no explicit mention of this step in the written description, and the preferred embodiment does not inherently include this step. The examiner notes, that upon further examination, compress is interpreted as, " to reduce the size of a set of data, such as a file or a communications message, so that it can be stored in less space or transmitted with less bandwidth." (Microsoft Computer Dictionary, Third Edition)

Claim Objections

14. Claims 1,2, and 17 are objected under 37 CFR 1.75(a) for being vague and indefinite.

Claim 1:

The phrase,"edge detecting valid instructions" is vague. It is unclear what is meant by edge detecting a valid instruction, which is merely data. Edge detecting normally occurs on signals. Take out the term "edge" if your intent is to detect valid instructions. The examiner notes, that upon further examination, "edge

detect valid instructions" is interpreted as using any edge detecting means on valid instructions.

Claim 2:

The phrase, "edge detecting valid instructions" is vague. It is unclear what is meant by edge detecting a valid instruction, which is merely data. Edge detecting normally occurs on signals. Take out the term "edge" if your intent is to detect valid instructions. The examiner notes, that upon further examination, "edge detect valid instructions" is interpreted as using any edge detecting means to determine valid instructions.

Claim 17:

The phrase, "edge detecting valid instructions" is vague. It is unclear what is meant by edge detecting a valid instruction, which is merely data. Edge detecting normally occurs on signals. Take out the term "edge" if your intent is to detect valid instructions. The examiner notes, that upon further examination, "edge detect valid instructions" is interpreted as using any edge detecting means on valid instructions.

15. Claims 1 – 17 are objected to because of the following informalities:

- Claims 1 – 17: The numbering of claims is not in accordance with 37 CFR 1.75(f) which requires the claims be numbered consecutively with Arabic numerals. Please take out the reference characters [c01] – [c17] and replace with Arabic numerals.

- Claim 3 is missing a period (.) at the end of the claim.

- Claim 16: The format to list a plurality of steps in the claim is incorrect in the phrase "the step of shifting the instructions occurs while at least one of i) the instructions occurring prior to the complex instruction are executed and ii) the complex instruction is executed" See MPEP 608.01(m) for the proper format to list a plurality of steps in a claim.

- Claim 17 should be "fetching a bundle" instead of "fetching an bundle".

16. Appropriate correction is required for all claim objections.

Claim Rejections - 35 USC § 112

The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

17. Claims 6 and 13 are rejected under 35 USC 112.

Claim 6:

18. The term "monotonic instruction set" is vague because it lacks antecedent basis in the description. Monotonic instruction set is not a known term in the art. Neither *Microsoft Computer Dictionary, Fifth Edition*, nor *The Authoritative Dictionary of IEEE Standards Terms, Seventh Edition*, include this term. One of ordinary skill in the art is not familiar with this term. The specification does not define a "monotonic instruction set". The examiner notes that this claim will not be further examined on its merits because the metes and bounds of the claim cannot be established since the meaning of "monotonic instruction set" cannot be determined.

Claim 13:

19. The term "monotonic instruction set" is vague because it lacks antecedent basis in the description. Monotonic instruction set is not a known term in the art. Neither Microsoft Computer Dictionary , Fifth Edition, nor The Authoritative Dictionary of IEEE Standards Terms, Seventh Edition, include this term. One of ordinary skill in the art is not familiar with this term. The specification does not define a "monotonic instruction set". The examiner notes that this claim will not be further examined on its merits because the metes and bounds of the claim cannot be established since the meaning of "monotonic instruction set" cannot be determined.

Claim Rejections - 35 USC § 102

20. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

21. Claim 1 is rejected under 35 U.S.C. 102(b) as being anticipated by Sato et al. (US Pat. 4,566,103).

Claim 1:

22. A method, comprising:

advancing instructions along a microprocessor pipeline; (Col. 2, lines 36 – 42) and

edge detecting valid instructions within the microprocessor pipeline. (Fig. 1,CS 3,

MIR 4, ECC 7, Fig. 3, ECC 7, FF 23/ Check 22 in Fig. 3 outputs an ECC error

signal, whose edge is detected by FF 23, in response to the validity of an

instruction (Col. 4, lines 61 – 68, col. 5, lines 1 – 4).)

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all

obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set
> forth in section 102 of this title, if the differences between the subject matter sought to be patented and
> the prior art are such that the subject matter as a whole would have been obvious at the time the
> invention was made to a person having ordinary skill in the art to which said subject matter pertains.
> Patentability shall not be negatived by the manner in which the invention was made.

23. Claims 1-5,7-12,14-17 are rejected under 35 U.S.C. 103(a) as being

unpatentable by Panwar et al. (US Pat. 6,098,165) in further view of Wakerly

(Third Edition, Digital Design, Principles & Practices).

Claim 1:

24. Panwar et al. disclose :

25. A method, comprising:

advancing instructions along a microprocessor pipeline; (Fig. 2, Col. 4, lines 62 –

67, Col. 5, line 1)

and

detecting valid instructions within the microprocessor pipeline. (Fig. 7, Valid

Vector 710/ The valid vector detects valid instructions in a sub-bundle. (Col. 11,

lines 22 - 23). This is done within the microprocessor pipeline (Fig. 7, Fig. 3, Control Logic 300, IRU 204, Fig. 2, IRU 204, Col. 11. lines 14 - 16))

26. Panwar et al. are silent about edge detecting valid instructions within the microprocessor pipeline, and only disclose detecting valid instructions within the microprocessor pipeline.

27. Wakerly teaches edge detecting in his description of edge-triggered behavior in the positive edge-triggered D flip flop. Figure 7 – 17. (Wakerly, Page 540 - 541, 7.5.2 Edge-triggered D Flip Flop, Figure 7 – 17.) The positive edge-triggered D flip flop captures the value of the input on the rising edge of the clock. The flip flop retains the value until the next rising edge of the clock.

28. One of ordinary skill in the art would have recognized that the valid vector comprises of data storage. The positive edge-triggered D flip flop is a type of data storage. The valid vector, implemented with positive edge-triggered D flip flops, captures the input value (Panwar et al., Fig. 7, XOR 706, Stall Vector 702 and Stall Vector 708) at the rising edge of the clock. By using positive edge-triggered D flip flops, the value of the valid vector is stable and constant until the next rising edge, providing adequate setup and hold times for sequential circuits that utilize this result, or a derivative of this result. Because there is adequate setup and hold times for the input signal to ensure a stable state, the logic in Panwar et al. is not subject to enter a metastable state due to a lack of setup and hold times from the valid vector.

29. One of ordinary skill in the art at the time of applicant's invention would have been motivated to implement the valid vector described by Panwar et al. with the positive edge-triggered D flip flops, described by Wakerly, to create stable data storage that provides adequate setup and hold times, which reduces the possibility of entering a metastable state for the logic that utilizes the result. The valid vector, implemented with positive edge-triggered D flips flops, would edge detect, on the rising edge of the clock, valid instructions, indicated by the XOR of the stall vectors in Fig. 7, within the microprocessor pipeline.

30. Therefore, it would have been obvious to implement the valid vector of Panwar et al. with the positive edge-triggered D flip flops described by Wakerly to obtain the invention as specified in claim 1.

Claim 2

31. Panwar et al. disclose:

32. A method, comprising:

fetching a bundle of instructions; (Col. 6, lines 4 – 5, 7 – 10) and

detecting valid instructions within the bundle (Col. 12, lines 14 – 35/ The valid vector indicates valid instructions in a sub-bundle within the bundle. (Col. 11, lines 22-23). The value of the valid vector is set by XOR between stall vector 708 and stall vector 702 (Col. 11, lines 40 - 41./ Fig. 7, Stall Vector 708, Stall Vector 702, and XOR 706) The valid vector detects the result of the XOR between the two input stall vectors to determine the valid instructions in the sub-bundle within the bundle.)

Application/Control Number: 10/010,389   Page 16
Art Unit: 2183

33. Panwar et al. are silent about edge detecting valid instructions within the bundle,
and only discloses detecting valid instructions within the bundle.

34. Wakerly teaches edge detecting in his description of edge-triggered behavior in
the positive edge-triggered D flip flop. Figure 7 – 17. (Wakerly, Page 540 - 541,
7.5.2 Edge-triggered D Flip Flop, Figure 7 – 17.) The positive edge-triggered D
flip flop captures the value of the input on the rising edge of the clock. The flip
flop retains the value until the next rising edge of the clock.

35. One of ordinary skill in the art would have recognized that the valid vector
comprises of data storage. The positive edge-triggered D flip flop is a type of
data storage. The valid vector, implemented with positive edge-triggered D flip
flops, captures the input value (Panwar et al., Fig. 7, XOR 706, Stall Vector 702
and Stall Vector 708) at the rising edge of the clock. By using positive edge-
triggered D flip flops, the value of the valid vector is stable and constant until the
next rising edge, providing adequate setup and hold times for sequential circuits
that utilize this result, or a derivative of this result. Because there is adequate
setup and hold times for the input signal to ensure a stable state, the logic in
Panwar et al. is not subject to enter a metastable state due to a lack of setup and
hold times from the valid vector.

36. One of ordinary skill in the art at the time of applicant's invention would have
been motivated to implement the valid vector described by Panwar et al. with the
positive edge-triggered D flip flops, described by Wakerly, to create stable data
storage that provides adequate setup and hold times, which reduces the

possibility of entering a metastable state for the logic that utilizes the result. The valid vector, implemented with positive edge-triggered D flips flops, would edge detect, on the rising edge of the clock, valid instructions, indicated by the XOR of the stall vectors in Fig. 7, in the sub-bundle within the bundle.

37. Therefore, it would have been obvious to implement the valid vector of Panwar et al. with the positive edge-triggered D flip flops described by Wakerly to obtain the invention as specified in claim 2.

Claim 3:

38. A method according to claim 2, further comprising shifting at least one instruction within the bundle. (Fig. 9/ Complex instruction I5 is one instruction within the bundle. When complex instruction I5 is converted into microinstructions, it is shifted from its position in the middle of the main bundle to the top of sub-bundle 4.)

Claim 4:

39. A method according to claim 3, further comprising rotating at least one instruction based at least in part on the number of valid instructions in the bundle. (Fig. 9/ Complex instruction I5 is one instruction within the bundle. When complex instruction I5 is converted into microinstructions, it is shifted from its position in the middle of the main bundle to the top of sub-bundle 4. The null entries that occupied the first two spaces are rotated to the end of the buffer.)

Claim 5:

40. A method according to claim 3, further comprising compressing the bundle of instructions. (Fig. 5, main bundle 203, sub-bundle 500A, 500B, 500C, Col. 10, lines 6 – 12/The instructions in the main bundle are compressed during their transmission to the processor. The compressing occurs as the instructions in the main bundle are parsed into sub-bundles and sent off for further processing at different times.)

Claim 7:

41. A method according to claim 3, further comprising compressing the bundle of instructions (Fig. 5, main bundle 203, sub-bundle 500A, 500B, 500C, Col. 10, lines 6 – 12/The instructions in the main bundle are compressed during their transmission to the processor. The compressing occurs as the instructions in the main bundle are parsed into sub-bundles and sent off for further processing at different times.)

based at least in part on the number of valid instructions in the bundle. (Fig. 6/ Col. 11, lines 22 – 23, 40 - 41).The size of the sub-bundles is dependent on the number of consecutive non-complex instructions in a main bundle. The non-complex instructions are marked as valid in the valid vector if the XOR evaluates to true, and thus considered valid instructions as well.)

Claim 8:

42. Panwar et al. disclose:

43. A method, comprising:

fetching a bundle of instructions having a complex instruction; (Col. 3, lines 49 –

54, Col. 6, lines 4 – 5, 7 – 10/ The IFU fetches a bundle of instructions.

Instructions are either complex or non-complex. Therefore, a bundle of

instructions is able to contain zero, one, ore more (up to the size of the bundle)

complex instructions.)

shifting at least one instruction occurring after the complex instruction; and (Fig.

9/ Complex instruction I5 is one instruction within the bundle. When complex

instruction I5 is converted into microinstructions, it is shifted from its position in

the middle of the main bundle to the top of sub-bundle 4. It occurs after the

complex instruction I2.)

 detecting the number of valid instructions occurring after the complex instruction.

(Col. 12, lines 14 – 35/ The valid vector indicates valid instructions in a sub-

bundle within the bundle. (Col. 11, lines 22 - 23). The value of the valid vector is

set by XOR between stall vector 708 and stall vector 702 (Col. 11, lines 40 - 41./

Fig. 7, Stall Vector 708, Stall Vector 702, and XOR 706) In the example (Col. 12,

lines 14 – 35), a complex instruction occurred at clk #2. The valid vector (Valid

Vector #3) for clk #3 detected the result of the XOR of the two input stall vectors

(Stall Vector #2, Stall Vector #3) to determine the number of valid instructions in

the sub-bundle occurring after the complex instruction.)

44. Panwar et al. is silent about edge detecting the number of valid instructions

occurring after the complex instruction, only discloses detecting the number of

valid instructions occurring after the complex instruction.

45. Wakerly teaches edge detecting in his description of edge-triggered behavior in the positive edge-triggered D flip flop. Figure 7 – 17. (Wakerly, Page 540 - 541, 7.5.2 Edge-triggered D Flip Flop, Figure 7 – 17.) The positive edge-triggered D flip flop captures the value of the input on the rising edge of the clock. The flip flop retains the value until the next rising edge of the clock.

46. One of ordinary skill in the art would have recognized that the valid vector comprises of data storage. The positive edge-triggered D flip flop is a type of data storage. The valid vector, implemented with positive edge-triggered D flip flops, captures the input value (Panwar et al., Fig. 7, XOR 706, Stall Vector 702 and Stall Vector 708) at the rising edge of the clock. By using positive edge-triggered D flip flops, the value of the valid vector is stable and constant until the next rising edge, providing adequate setup and hold times for sequential circuits that utilize this result, or a derivative of this result. Because there is adequate setup and hold times for the input signal to ensure a stable state, the logic in Panwar et al. is not subject to enter a metastable state due to a lack of setup and hold times from the valid vector.

47. One of ordinary skill in the art at the time of applicant's invention would have been motivated to implement the valid vector described by Panwar et al. with the positive edge-triggered D flip flops, described by Wakerly, to create stable data storage that provides adequate setup and hold times, which reduces the possibility of entering a metastable state for the logic that utilizes the result. The valid vector, implemented with positive edge-triggered D flips flops, would edge

detect, on the rising edge of the clock, the number of valid instructions, illustrated

by the valid vector, occurring after the complex instruction.

48. Therefore, it would have been obvious to implement the valid vector of Panwar et

al. with the positive edge-triggered D flip flops described by Wakerly to obtain the

invention as specified in claim 8.

Claim 12:

49. A method according to claim 8, wherein the step of shifting the instructions

comprises compressing the instructions occurring after the complex instruction.

(Fig. 5, main bundle 203, sub-bundle 500A, 500B, 500C, Fig. 9, Col. 10, lines 6 –

12/The instructions in the main bundle are compressed during their transmission

to the processor. The compressing occurs as the instructions in the main bundle

are parsed into sub-bundles and sent off for further processing at different times.

The complex instruction is I2 in Fig. 9. The non-complex instructions which occur

after the complex instruction, I3 and I4, are compressed as they are placed in

sub-bundle 3 and sent down the pipeline for further processing once another

complex instruction, I5, is detected and converted to microinstructions, which

encompass the step of shifting.)

Claim 14:

50. A method according to claim 8, further comprising executing instructions

occurring prior to the complex instruction during a first clock cycle. (Col. 2, lines

35 – 38, Col. 6, lines 59 – 64, Col. 7, lines 23 – 24, Col. 10, lines 50 – 53, Fig. 6,

element 604/ The instructions occurring prior to the complex instruction are

placed in a sub-bundle. Once a complex instruction is detected, then the instructions are issued for execution. This occurs during a clock cycle because the processor is synchronous.)

Claim 15:

51. A method according to claim 14, further comprising executing the complex instruction during a second clock cycle. (Fig. 6, element 606, 608, and 610, Col. 8, lines 48 – 50./ After the complex instruction is expanded during a clock cycle, it is issued for execution during another, or second, clock cycle. This clock cycle is different from the clock cycle where the non-complex instructions in the sub-bundle are issued for execution. (See Fig. 6, element 604, 606)).

Claim 16:

52. A method according to claim 15, wherein the step of shifting the instructions occurs while at least one i) the instructions occurring prior to the complex instruction are executed (Fig. 6, elements 604, 606, Col. 8, lines 48 – 50/ The shifting of instructions occurs when the complex instruction is converted into microinstructions (Fig. 6, element 606). This conversion takes at least one clock cycle. The instructions occurring prior to the complex instruction are scheduled for execution (Fig. 6, element 604) as opposed to being directly issued. Therefore, the instructions occurring prior to the complex instruction are executed while the complex instruction is expanded into microinstructions.)

and ii) the complex instruction is executed.

Claim 17:

53. Panwar et al. discloses :

54. A method, comprising:

fetching a bundle of instructions having a complex instruction; (The IFU

fetches a bundle of instructions. (Col. 6, lines 4 –5, 7 – 10). Instructions are either

complex or non-complex. (Col. 3, lines 49 – 54). Therefore, a bundle of

instructions is able to contain zero, one, or more (up to the size of the bundle)

complex instructions.)

executing during a first clock cycle valid instructions occurring prior to the

complex instruction; (Fig. 6/ The instructions occurring prior to the complex

instruction are placed in a sub-bundle, and determined valid by the valid vector.

(Col. 12, lines 23 – 25) Once a complex instruction is detected, then the

instructions are issued for execution. (Fig. 6, element 604/ Col. 10, lines 50 - 53)

Since the IEU pipeline is one or more stages, the execution takes one or more

cycles. (Col. 7, lines 23 – 24) The instructions are executed during a clock cycle

(known as the first clock cycle) because the processor is synchronous. (Col. 2,

lines 35 – 38). In the example (Col. 12, lines 14 – 25), these are the instructions

that pass through during clk #1.)

executing the complex instruction during a second clock cycle; (Fig. 2, ISU

206, FGU 210, and IEU 208, fig. 6, element 606, 610, Col. 7, lines 23 – 56/After

the complex instruction is issued for execution (Fig. 6, element 610), the

execution of the complex instruction itself takes one or more clock cycles. In the

example (Col. 12, lines 14 – 35), this is the complex instruction that occurs at clk

#2. An arbitrary clock cycle that occurs while the complex instruction is executing which is not the first clock cycle is the second clock cycle.)

shifting instructions occurring after the complex instruction during at least one of the first clock cycle and the second clock cycle; (Fig. 9, Col. 7, lines 23 – 56, Col. 12, lines 14 – 35/ As illustrated in Fig. 9, complex instruction I5 is one instruction within the bundle after complex instruction I2. When complex instruction I5 is converted into microinstructions, it is shifted from its position in the middle of the main bundle to the top of sub-bundle 4. In the example (Col. 12, lines 14 – 35), a complex instruction occurred at clk #2, and the instructions in question are passed at clk #4. Since the IEU pipeline has one or more stages, the shifting of the instructions occurs while the complex instruction is executing and during the second cycle.)

detecting valid instructions occurring after the complex instruction during at least one of the first clock cycle and the second clock cycle; and (Col. 7, lines 23 – 26, Col. 12, lines 14 – 35/ The valid vector indicates valid instructions in a sub-bundle within the bundle. (Col. 11, lines 22 - 23). The value of the valid vector is set by XOR between stall vector 708 and stall vector 702 (Col. 11, lines 40 - 41./ Fig. 7, Stall Vector 708, Stall Vector 702, and XOR 706) In the example (Col. 12, lines 14 – 35), a complex instruction occurred at clk #2. The valid vector (Valid Vector #3) for clk #3 detected the result of the XOR of the two input stall vectors (Stall Vector #2, Stall Vector #3) to determine the valid instructions in the sub-bundle occurring after the complex instruction. This occurs during the first

cycle when the non-complex instructions are executing since the IEU pipeline is
one or more stages.)

executing the valid instructions occurring after the complex instruction
during a third clock cycle (Col. 7, lines 23 – 26, Col. 12, lines 14 – 35/ The valid
instructions occurring after the complex instruction are the non-complex
instructions passed through at clk #3. Since the IEU pipeline is one or more
stages, they are executing during a third clock cycle which is not the first clock
cycle or second clock cycle.)

55. Panwar et al. is silent about edge detecting valid instructions occurring after the
complex instruction during at least one of the first clock cycle and the second
clock cycle, and discloses only detecting valid instructions occurring after the
complex instruction during at least one of the first clock cycle and the second
clock cycle.

56. Wakerly teaches edge detecting in his description of edge-triggered behavior in
the positive edge-triggered D flip flop. Figure 7 – 17. (Wakerly, Page 540, 7.5.2
Edge-triggered D Flip Flop, Figure 7 – 17.) The positive edge-triggered D flip flop
captures the value of the input on the rising edge of the clock. The flip flop retains
the value until the next rising edge of the clock.

57. One of ordinary skill in the art would have recognized that the valid vector
comprises of data storage. The positive edge-triggered D flip flop is a type of
data storage. The valid vector, implemented with positive edge-triggered D flip
flops, captures the input value (Panwar et al., Fig. 7, XOR 706, Stall Vector 702

and Stall Vector 708) at the rising edge of the clock. By using positive edge-triggered D flip flops, the value of the valid vector is stable and constant until the next rising edge, providing adequate setup and hold times for sequential circuits that utilize this result, or a derivative of this result. Because there is adequate setup and hold times for the input signal to ensure a stable state, the logic in Panwar et al. is not subject to enter a metastable state due to a lack of setup and hold times from the valid vector.

58. One of ordinary skill in the art at the time of applicant's invention would have been motivated to implement the valid vector described by Panwar et al. with the positive edge-triggered D flip flops, described by Wakerly, to create stable data storage that provides adequate setup and hold times, which reduces the possibility of entering a metastable state for the logic that utilizes the result. The valid vector, implemented with positive edge-triggered D flips flops, would edge detect, on the rising edge of the clock, valid instructions, indicated by the XOR of the stall vectors in Fig. 7, occurring after the complex instruction during at least one of the first clock cycle and the second clock cycle.

59. Therefore, it would have been obvious to implement the valid vector of Panwar et al. with the positive edge-triggered D flip flops described by Wakerly to obtain the invention as specified in claim 17.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Latha Ravindran whose telephone number is (703)305-

8115. The examiner can normally be reached on Monday through Friday 8:30am to 5:00pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (703) 305-9712. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Latha Ravindran
Examiner
Art Unit 2183

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100